-1-

Date: 10|26|01   Express Mail Label No. EV 010604045 US

Inventor: Nabil A. Abu El Ata

Attorney's Docket No.:   3023.1001-004

SYSTEM AND METHOD FOR IMPROVING PREDICTIVE MODELING OF AN INFORMATION SYSTEM

RELATED APPLICATIONS

This application is a continuation-in-part of Application No. 09/127,191, filed
5   July 31, 1998, which claims the benefit of U.S. Provisional Application No. 60/085,350,
filed on May 13, 1998. The entire teachings of the above applications are incorporated
herein by reference.

BACKGROUND OF THE INVENTION

10   With the advent of electronic computing, business organizations have deployed
computerized information systems to provide time-critical, cost-efficient business
solutions. Information systems typically include various software applications
distributed across one or more hardware/network operating environments.

In developing such systems, traditional system engineering involves multiple
15   development phases, including a requirements phase, an architecture design phase, a
construction phase, and a deployment phase. During the design phases, static
descriptions and assumptions about hardware and software component behavior and
characteristics are relied on for developing the system architecture of the information
system.

20   However, in deployed systems, the characteristics and behavior associated with
individual hardware and software components are dynamic. Thus, the information
system as a whole is also associated with dynamic characteristics and behavior.

Changes in workload and hardware and software interactions typically have a significant effect on system performance.

With traditional system engineering, dynamic characteristics and behavior are not addressed until late in the development process, if at all, where the improvement
5   possibilities are more limited. Thus, there is no guarantee that an information system, once deployed, will satisfy current and future business requirements, such as business-critical response time and throughput.

Furthermore, problem isolation and debugging becomes more complicated, resulting in increased development costs and time. In particular, if the origin of a
10   problem resides in the business or architecture design itself, the cost of improvement may become prohibitive without partial or full redesign. Thus, with traditional system engineering, it is difficult, if not impossible, to guarantee the deployment of complex business information systems within time and budget constraints having required performance and operating costs.

15   SUMMARY OF THE INVENTION

Embodiments of the invention provide a system and method for improving predictive modeling of an information system. According to one embodiment, a business solution is modeled with a dynamic representation. The system includes an input module, a construction module, a performance metric calculation module, and an
20   output module. The input module provides a description of a business solution to the construction module with the description of the business solution describing business components and interactions among the business components. According to one embodiment, the business components include business processes or sub-processes, business functions or sub-functions, and data stores. The construction module, in turn,
25   generates a predictive model of the information system, including a business layer generated from the business solution description. The business layer models the dynamic characteristics and behavior of the business components and the interactions

among the business components in response to dynamic business workloads, such that a dynamic representation of the business solution results.

The construction module may also generate an application layer and a system layer of the predictive model, expressing dynamic characteristics and behavior of

5   corresponding application and system components that support the business components and their interactions. A performance metric calculation module, in turn, calculates performance metrics from the predictive model for each layer as a function of the modeled dynamic characteristics and behavior. The resulting performance metrics of the business layer indicate whether the business solution satisfies a set of business

10   requirements regardless of whether the performance metrics of the application and system layers are acceptable.

Further embodiments provide a system and method for improving the accuracy of a predictive model of an information system through automated calibration of the predictive model against predefined performance benchmarks. In particular, a

15   construction module generates a predictive model of an information system including a business layer, an application layer, and a system layer with each layer modeling dynamic characteristics and behavior of one or more components. A performance metric calculation module calculates individual performance metrics for each component modeled in the application and system layers from the dynamic

20   characteristics and behavior. The construction module, in turn, compares the calculated individual performance metrics against predefined individual performance benchmarks to assess the accuracy of the predictive model. For component models that do not substantially match a corresponding performance benchmark, the construction module performs a sensitivity analysis on individual component models.

25   According to one embodiment, a sensitivity analysis includes the construction module adjusting one or more parameters of a model equation that expresses the dynamic characteristics and behavior of a component model. The performance metric calculation module then calculates updated performance metrics for each component model in the application and system layers with the construction module comparing the

individual performance metrics against individual performance benchmarks to assess the accuracy of the predictive model again. This process repeats until all of the individual performance metrics are within a predefined threshold of the individual performance benchmarks.

5   BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, features and advantages of the invention will be apparent from the following more particular description of preferred embodiments of the invention, as illustrated in the accompanying drawings in which like reference characters refer to the same parts throughout the different views. The drawings are not

10   necessarily to scale, emphasis instead being placed upon illustrating the principles of the invention.

FIG. 1 is a diagram of a simple information system implementing a business solution according to one embodiment of the present invention.

FIG. 2 is a diagram illustrating the components of a system that generates a

15   multi-layer predictive model according to one embodiment of the present invention.

FIG. 3 is a conceptual diagram of a multi-layer predictive model according to one embodiment of the present invention.

FIG. 4 is a flow diagram of multi-layer predictive modeling according to one embodiment of the present invention.

20   FIG. 5 is a flow diagram illustrating a sequence of development phases interleaved with a predictive modeling phase according to one embodiment of the present invention.

FIG. 6 is a flow diagram illustrating a process for validating a proposed information system design resulting from a design phase according to one embodiment of the present invention.

5   FIG. 7 is a diagram illustrating a conceptual business process design according to one embodiment of the present invention.

FIG. 8 is a diagram illustrating a refined business process design according to one embodiment of the present invention.

FIG. 9 is a diagram illustrating a technical architecture design according to one
10   embodiment of the present invention.

FIG. 10 is a flow diagram of a process for validating a prototype of an information system from a construction phase according to one embodiment of the present invention.

15   FIG. 11 is a diagram illustrating a prior art representation of a business process design for a stock exchange.

FIG. 12 is a diagram illustrating a dynamic business representation of a business solution according to one embodiment of the present invention.

FIG. 13 is a flow diagram of a process for improving the accuracy of predictive
20   modeling according to one embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

A description of preferred embodiments of the invention follows.

Business solutions involve interactions among business components, including business workloads, business processes, and data stores, to solve the needs of business entities. Generally, the design of such information systems are constrained by a set of business requirements, which predefine certain performance criteria to make the

5 business solution acceptable. Information systems implement business solutions by providing a technical infrastructure that supports the business workload, business processes, and data storage requirements of the solution.

FIG. 1 is a diagram of a simple information system implementing a business solution according to one embodiment of the present invention. An information system

10 typically includes a system architecture with software application components distributed across system hardware and system networking components. Referring to FIG. 1, the information system includes client computers, c_1 through c_n, executing a data processing client application 15 for implementing the input and output of business workload 10; a server, SVR, executing a data processing server application 25 for

15 implementing a business process 20; and a data store, DS, executing a data based management application 35 for implementing data storage functionality 30. However, most information systems are much more complex, including a large number of distributed applications, client terminals, servers, data stores, internetworking infrastructure, and a variety of networked peripheral devices.

20 In designing and implementing information systems, traditional system engineering typically proceeds through several development phases from conception through deployment. However, there are no checkpoints to determine whether the design or implementation will satisfy a set of predefined business or technical performance criteria. Without such predictive assessment, a significant amount of time

25 and investment may be wasted in developing information systems that may not be able to satisfy the business requirements within time and budget constraints.

For example, with respect to FIG. 1, it is difficult, if not impossible, to guarantee the response time and throughput of this design. Depending on the expected business workload, this design may need additional server capacity to satisfy its business and

performance requirements, thereby increasing the cost of development. If such design and implementation modifications are realized earlier in the development process, a significant amount of time and investment may be saved.

Embodiments of the present invention provide a system and method for
5   multi-phased system development of information systems utilizing predictive modeling to validate the design and construction of an information system at each phase of development. Such embodiments provide early detection of unacceptable designs and implementations early in the development lifecycle, avoiding significant losses in investment. According to one embodiment, predictive modeling may be implemented
10   as described in U.S. Patent Application Serial No. 09/127,191, filed July 31, 1998, entitled "Method and Apparatus for Designing and Analyzing Information Systems Using Multi-Layer Mathematical Models." The entire contents and teachings of which are incorporated herein by reference. However, further embodiments of predictive modeling known to those skilled in the art may also be employed.

15   Further embodiments of the invention provide a system and method for improving the accuracy of predictive modeling of an information system by modeling a dynamic representation of the business solution and through automated calibration of a predictive model against predefined performance benchmarks. With improved predictive modeling capacity, confidence may be instilled in a particular design or
20   implementation.

FIG. 2 is a diagram illustrating the components of a system that generates a multi-layer predictive model according to one embodiment of the present invention. The information system 40 may include an input module 46, a construction module 48, a performance metric calculation module 54, and an output module 56.

25   The input module 44 receives input 42 from an input device, a network, or a storage device. The input 42 includes a description for a proposed information system in varying degree of detail. In one embodiment, the input 42 is descriptive input that provides a complete description of the business processes within the organization, and is not limited to computer transactions or processes. In another embodiment, the input 42

also includes non-computer transactions such as paper transactions that do not occur on a computer, and even verbal transactions or processes that occur between people.

Generally, the input module 46 passes on data to the construction module 48, and the data is processed by the construction module 48, resulting in a predictive model

5   50 of the information system. The predictive model 50 is a quantitative model of the proposed information system created by the construction module 48 based on the descriptive input 42. The construction module 48 then passes the model 50 on to the performance metric calculation module 54 for further processing and then to the output module 56. The output module 56 provides output 58 to an output device, a network, or

10 a storage device. In one embodiment, the output module provides the output 58 to a display device for the designer of the information design system 40. For more information regarding multi-layer predictive modeling, refer to U.S. Patent Application Serial No. 09/127,191, filed July 31, 1998, entitled "Method and Apparatus for Designing and Analyzing Information Systems Using Multi-Layer Mathematical

15 Models." The entire contents and teachings of which are incorporated herein by reference.

FIG. 3 is a conceptual diagram of a multi-layer predictive model according to one embodiment of the present invention. Using a combination of deterministic and probabilistic mathematics, a multi-layer predictive model expresses the dynamic

20 characteristics and behavior of a proposed information system from three or more perspectives, including a business layer 60, an application layer 70, a system layer 80, and, optionally, a data layer (not shown). Through a system of equations, each layer models the dynamic characteristics and behavior of its components individually and collectively in terms of probabilities for delays, conflicts, contentions, and locks. Each

25 layer may have an effect on the dynamic characteristics and behavior expressed in the other layers, as indicated by arrows 65 and 75. Each layer may include further sublayers to provide additional levels of granularity to the predictive model.

According to one embodiment, the business layer 60 models the dynamic characteristics and behavior of business processes, data stores, and business I/O

workloads. The application layer 70, in turn, models the dynamic characteristics and behavior of application software components supporting the business components of the business layer 60. Information from the business layer 60, such as business workload, may affect the expression of the models in the application layer 70. For example, a

5   single interaction between business components in the business layer 60 may correspond to two requests and responses between corresponding application components in the application layer 70.

The system layer 80, in turn, models the dynamic characteristics and behavior of hardware and network components that provide a system infrastructure for distributing

10   the application software components of the application layer 70. Information from the application layer, such as application workload, may affect the expression of the models in the system layer 80. For example, an application request in the application layer 70 may correspond to 4 CPU and 20 I/O transactions in the system layer 80.

With a multi-layer predictive model of an information system, performance

15   metrics may be generated for each component, for each layer, and for the information system in general from the probabilities calculated from the predictive model for delays, conflicts, contentions, and locks.

FIG. 4 is a flow diagram of multi-layer predictive modeling according to one embodiment of the present invention. According to one embodiment, one or more sets

20   of model input parameters 110 are derived by the input module 46 (FIG. 2) from a description of an information system design. The model input parameters 110, in turn, are used by a construction module 48 in order to generate the predictive model 50 as a system of equations 120 representing the system, layer, and component probabilities related to a dynamic characteristic or behavior. A performance metric calculation

25   module 54 calculates performance metrics, referred to as Validation Output 125, using the probabilities calculated from the predictive model 50.

According to one embodiment, the system of equations 120 is expressed as follows:

$$P = \sum_i A_i\, P_i(S_i)\, \prod C_{j,m} P_{j,m}(S_{j,m}) \tag{1}$$

As previously stated, the system of equations 120 is used to generate

5   probabilities associated with a particular performance metric, such as response time and throughput.

Referring to equation (1), P is the general probability associated with the overall information system for a particular performance metric. The probability P is calculated from the sum of the probabilities $P_i(S_i)$ calculated for each layer i (*e.g.*, 1=business

10   layer 60, 2=application layer 70, 3=system layer 80). For each layer i, the probability $P_i(S_i)$ is calculated from the product of the probabilities $P_{j,m}(S_{j,m})$ calculated for each component in that layer. $P_{j,m}(S_{j,m})$ is a mathematical expression of any form expressing the probability of the performance metric occurring within a component.

With respect to the probability equation $P_{j,m}(S_{j,m})$ for a component, subscript 'j'

15   identifies the component, while subscript 'm' represents the states of services provided by a component. For example, a component j can be operated in one or more modes of operation (*e.g.*, batch, transactional, query-based), the value of 'm' indicates the mode of operation in use. Each equation $P_{j,m}(S_{j,m})$ may include terms which may be enabled and disabled by the value of 'm'.

20   A_i is a parameter representing the total workload associated with layer i and may depend on the workload associated with another layer (*e.g.*, A_2 being a function of A_1). Similarly, $C_{j,m}$ is a parameter representing the workload associated with component j. The value of $C_{j,m}$ may be a percentage of the workload A_i. For example, if A_i is equal to 100 and represents the total number of stock-related requests, workload $C_{1,m}$ to a seller

25   handling business process may be equal to 40% of A_i, representing the dynamic number of seller stock requests. Similarly, workload $C_{2,m}$ to a buyer handling business process may be equal to 60% of A_i, representing the dynamic number of buyer stock requests. It may also take into account data from one or more of the previous layers. Parameters A_i

and $C_{j,m}$ may be adjusted during model calibration, which is described in more detail with reference to FIG. 13. The solution of the system of equations will determine all values of these constants that, in turn, are used to calculate the modeled output performance metrics.

5   This three level modeling of FIGS. 3 and 4 allows the computation of both the service time and response time for a path as well as the service time and response time spent in each component on the path. The two metrics will permit the assessment of the performance potential of a design by assessing each of them differently. The service time translates the residence time of the path if no wait is exercised (i.e. no contention,

10   or conflict for resources). If such time is unacceptable, a re-engineering of the architecture or another design could become the only way to improve the situation. If the service time on the other hand is acceptable but the response time is not, this means that the waiting portion of the response time is unacceptable and only an optimization process might be required, including hardware upgrade. With such approach, one might

15   use this process to determine if an architecture or a design will operationally be able to fulfill the business need or another alternative might exist to improve the implementation while it is still possible, and definitely prior to any investment. The architecture and design improvement process of the present invention will be simply performed through the computation of different scenarios of change of the original

20   model until satisfactory values of performance metrics are obtained.

FIG. 5 is a flow diagram illustrating a sequence of development phases interleaved with a predictive modeling phase according to one embodiment of the present invention. The system development phases, as illustrated, include one or more requirements phases 210, one or more business process design phases 220, one or more

25   technical architecture design phases 230, one or more construction phases 240, one or more deployment phases 250, and one or more predictive modeling phases 260, which follow the techniques of FIGS. 3 and 4 discussed above.

During the predictive modeling phase 260, the design or implementation of an information system resulting from a development phase 220, 230, 240, 250 is validated

prior to proceeding to further development phases. Thus, predictive modeling ensures that the designs resulting from the design phases satisfy a predefined set of business or technical requirements. Likewise, predictive modeling ensures that the implementations of the information system or portions thereof conform to modeled characteristics and

5 behavior. If a design or implementation is not satisfactorily validated during the predictive modeling phase 260, the design or implementation is modified addressing the problems which prevent it from being validated. Thus, system behavior and performance is known at each phase of development prior to and through deployment of the information system.

10 During the requirements phase 210, requirements for the business solution are obtained from a variety of sources, including organizational departments within a business entity and its customers. These requirements define the criteria for successful implementation of a business solution, such as business-critical response times and throughput. Furthermore, these requirements express the business characteristics,

15 drivers, and constraints, driving the need and design of a business solution, enhancement, or replacement.

According to one embodiment, the design phases may include a business concept design phase 222, a business refinement design phase 224, and a technical architecture design phase 230. By validating the designs resulting from each phase

20 through predictive modeling, a thorough understanding of the capabilities of the entire information system is achieved and adjustments may be incorporated before investing significantly in an unacceptable design.

FIG. 6 is a flow diagram illustrating a process for validating a proposed information system design resulting from a design phase according to one embodiment

25 of the present invention.

At 310, a description of components and interactions involved in a proposed information system design is provided. According to one embodiment, the proposed design may be input using a Unified Modeling Language (UML) tool, such as Rational

Rose®, or another tool of like capability and exported into a format describing the design.

At 320, a predictive model 50 is generated from the description. According to one embodiment, the description is converted into a set of model input parameters 110 that are used to generate a system of equations 120 representing the dynamic characteristics and behaviors of components individually and collectively through one or more layers 60, 70, 80 of the predictive model 50. The values of the model input parameters 110 may be used for populating the equations as parameters, selecting model equations, or enabling terms within a model equation.

At 330, a performance metric calculation module 54 calculates performance metrics 125 from the predictive model 50. In particular, the performance metric calculation module 54 solves the system of equations 120 from the predictive model 50, resulting in probabilities for dynamic characteristics and behavior, such as delays, conflicts, constraints, and contentions. These probabilities, in turn, are utilized in calculations for various performance metrics, such as those described in more detail in U.S. Patent Application No. 09/127,191. The performance metrics may be calculated individually or collectively and are output at 125 in FIG. 4.

At 340, the performance metrics 125 calculated from the predictive model 50 are compared against a set of business or performance requirements illustrated as Validation Input 115 (FIG. 4). According to one embodiment, the comparison is a simple difference operation. Other comparators 130 are suitable.

At 350, if the calculated performance metrics 125 from the predictive model 50 satisfy the requirements 115, then the design at this design phase is validated and may proceed to a next phase of development at 360. Alternatively, the validated design may be further analyzed from a cost perspective to determine whether the design is profitable before proceeding to a next phase.

Conversely, at 350, if the modeled and calculated performance metrics 125 do not satisfy the set of performance requirements 115, the process proceeds to 370 where the design is modified addressing the problems preventing the design from validation.

For example, if the instant design is unable to handle an expected workload within an acceptable response time, additional capacity (*i.e.*, number of business, application, and/or system components) may be needed. Likewise, one or more components may be substituted in the design of the information system with different kinds of components

5   that may have more appropriate dynamic characteristics and behavior.

In some instances, the components causing negative performance results are provided additional services, such as security, reliability, modifiability, serviceability, and portability, enhancing the quality and robustness of the information system. Thus, the calculated performance metrics 125 may be used as an indicator to evaluate tradeoffs

10   involved in maintaining such services. If having such services is more important than performance, then the design may be acceptable even if the system performs at a lesser efficiency.

Once the design is modified at 370, the process returns back to 310 where an updated description of the proposed information system is provided for the validation

15   process of FIG. 6 to validate the design again.

In more detail with respect to design validation through predictive modeling, FIG. 7 is a diagram illustrating a conceptual business process design according to one embodiment of the present invention. A conceptual business process design includes high level definitions of business components (*e.g.*, business processes, business

20   workload, data stores) and their interactions.

According to one embodiment, the high level definition for a business workload may include (i) workload style (*e.g.*, electronic file transmission, electronic tape transmission, interactive computer I/O, facsimile transmissions, etc.); (ii) arrival rate (*i.e.*, frequency); (iii) destination (*e.g.*, business process or data store); and (iv) size per

25   unit workload. The high level definition for a business process may include (i) interactions with business components; (ii) the order in which the interactions are executed; (iii) frequency of each interaction; (iv) message sizes initiating each interaction; and (v) interaction style (*e.g.*, unidirectional or bidirectional, serial or

parallel, electronic or manual). The high level definition for a data store may include the size of the data store.

Generally, there are not many specifics known about the technical architecture at this phase, if any. Thus, during the predictive modeling phase 260, the application and system layers 70, 80 are populated with standard component models provided by a component library. According to one embodiment, a user interface is provided through which each business component may be mapped to a business application, modeled by one or more software component models in the application layer 70. Each business application, in turn, is mapped to a default hardware operating environment in the system layer 80, which includes a set of hardware and network component models. Thus, the resulting multi-layer predictive model 50 may be used to predict rough estimates of performance metrics, which may be used to determine the viability of the business concept.

For example, the arrival rates of business inputs (e.g., number of transactions per day) and outputs can be fed through the predictive modeling process to derive the response time that would be expected from the business solution, when handling that volume. If the performance metrics indicate that the business solution is able to handle the expected throughput within the required response time, the design of the business concept is viable, warranting additional investment. If performance metrics indicate that the business solution is able to handle the expected throughput, but not within the required response time, then the conceptual business process design may need to be modified. In particular, the predictive model 50 can be used to determine if additional capacity in the form of additional business processes working in parallel may accomplish the given task in a more acceptable window of time. Based on these projections, the cost to deploy the business solution, in a configuration with acceptable response time and throughput, is used to define the business case for the proposed solution, such as how to make it a profitable proposition.

System development iterates between the predictive modeling phase 260 and the business concept design phase 222 until the performance metrics of the business layer

60 either satisfy the business requirements or indicate that the business solution is not viable. If it is determined that the design is not viable, the business concept may need to be re-designed or discarded. Thus, the utilization of predictive modeling at this preliminary phase prevents further investment losses.

5   FIG. 8 is a diagram illustrating a refined business process design according to one embodiment of the present invention. During the business refinement design phase, the conceptual business process design, validated from the business concept design phase 222, is the basis for a refined business process design, providing an additional level of granularity to the definitions of the conceptual business process design.

10   In particular, general business processes are broken out into elementary business processes allowing for specialized processing and avoiding duplication of common processes. Data stores are further defined, identifying table entities and their interactions with the elementary business processes. Business workloads may be newly defined or redefined according to the refined business process design. Referring to

15 FIG. 8, business process BP1 is replaced by two elementary business processes, EBP1 and EBP2; Data Store A is further defined describing three table entities (*i.e.* FIG. 8, Entity 1; Entity 2; Entity 3); and business workload Output is broken out into two outputs, Output A and Output B (FIG. 8). Furthermore, interactions between the business components may be added or redefined according to the refined business

20 process design.

According to one embodiment, the definitions for the business components are similar to those defined during the business concept design phase 222, with the exception that the definitions may be more accurate at this phase. However, the definition of a data store may additionally include the identity, size, and depth of table

25 entities and usage.

Generally, there are not many specifics known about the technical architecture at this phase, if any. Thus, during the predictive modeling phase 260, the application and system layers 70, 80 are populated with standard component models provided by a

component library as previously described in connection with conceptual business process designs.

However, according to one embodiment, the predictive performance may be computed across specific target platforms modeled through system component models

5   in the system layer 80 to allow capacity planning for all components of the system and to refine the business costs, cash flow/capital outlays to be projected. Thus, the resulting multi-layer predictive model may be used to project rough estimates of performance metrics, which may be used to determine the viability of the refined business processes.

10   Due to the increased complexity of the refined business process design, conflicts, contentions, and locks may emerge among elementary business processes utilizing the same resources. The performance metrics 125 calculated from the predictive model 50 may indicate such conflicts with increased business response times, which are caused by delays associated with the conflicts. As with the business concept

15   design phase 222, the performance metrics may be utilized to isolate and remedy the origin of such conflicts, contentions, and locks.

System development iterates between the predictive modeling phase 260 and the business refinement design phase 224 until the performance metrics 125 of the business layer 60 either satisfy the business requirements 115 or the business solution is

20   determined to be not viable. This is illustrated as accuracy evaluation 140 in FIG. 4.

FIG. 9 is a diagram illustrating a technical architecture design according to one embodiment of the present invention. During the technical architecture design phase 230, the business process design, validated from the business refinement design phase 224, is the basis for formulating a proposed design for the technical architecture design

25   of an information system. A technical architecture design includes definitions of both hardware and software components (*e.g.*, technical processes, application programs, control data, hardware operating environments) and their interactions.

The technical architecture design breaks down the elementary business processes of FIG. 8 into descriptions of technical processes, representing business applications.

Each technical process is further decomposed with definitions describing software programs that implement the individual functionality provided by the technical process. Control data may be provided to the software programs through data files or more sophisticated data registries. The actual sizes for message flows between these

5 programs, their frequency, and style (*e.g.*, message-based transfer) may be described. Descriptions of the programs may include code or pseudo code segments that may be utilized during one or more of the construction phases.

Thus, during the predictive modeling phase 260 at this juncture, the application and system layers 70, 80 are populated with either standard or customized component

10 models. According to one embodiment, the construction module 48 may either propose a standard component model from the component library or guide the user through additional configuration screens to generate a customized component model. Thus, the resulting multi-layer predictive model 50 may be used to predict accurate values of performance metrics, which may be used to determine the viability as well as actual

15 characteristics and behavior of the information system.

For example, through predictive modeling, the maximum business arrival rate that the solution can support is known as well as the response time that can be achieved. Predictive modeling may also be utilized to determine the location of system bottlenecks and the threshold at which each component reaches full capacity or

20 performance limit. This information can lead to reconfiguration options that will allow the system to be further scaled.

Turning to the one or more construction phases 240 of FIG. 5, each construction phase 240 may result in prototypes of at least a portion of an information system constructed from a validated design. By validating the implementation of the prototypes

25 through predictive modeling, the resulting system can be deployed with reasonable confidence that it will perform as expected. Furthermore, by comparing these estimates with measured results in the test environment, weaknesses within the operating environment may be derived. Thus, preventative or corrective action may be taken in anticipation of components not running as efficiently as they should. Refinement of the

physical solution with the model allows the business to have optimal performance from the solution, yielding the best cost/performance ratio for the solution and the best return on investment.

FIG. 10 is a flow diagram of a process for validating a prototype of an information system from a construction phase 240 according to one embodiment of the present invention. In FIG. 4, this is referred to as a stability assessment 150 for prototypes of portions of the information system and a representative assessment 160 for prototypes of the entire information system.

At 410, a prototype of at least a portion of the information system is constructed from a validated design, typically resulting from the technical architecture design phase 230. Physical construction often modifies some specifics in a design in which components are substituted due to unavailability, nonconformance to published specifications, expense, or undesirable latent characteristics and behavior.

At 420, the predictive model 50 of the validated design is updated with changes or substitutions incorporated during the construction of a prototype. In particular, components within the predictive model are substituted for other standard or customized component models.

At 430, individual or group performance metrics 125 are calculated from the predictive model 50 for different kinds and/or volume of workload.

At 440, actual performance metrics are obtained from the prototype in response to the different kinds and/or volume of workload.

At 450, the actual performance metrics are compared against the modeled performance metrics 125 to verify that the prototype conforms to the predictive model 50.

If, at 460, the actual performance metrics match the modeled performance metrics 125 within a predefined threshold, the implementation of the prototype conforms to the predictive model 50 and the process may proceed to a next development phase at 470. Alternatively, if, at 460, the actual performance metrics do not match the modeled performance metrics 125, then the prototype implementation does not conform

to the predictive model and the process proceeds to 480 where the implementation of the prototype is reevaluated and modified. Thereafter, the modified/revised prototype is validated by reiterating through steps 420, 430, 440, 450, and 460.

Finally, the system may be deployed with confidence that it will perform as
5   expected. The system may be further monitored in production for information to add to the predictive model 50, thus, providing early warning of changes in performance, throughput scalability, or capacity requirements. If the implementation of a prototype is not validated, adjustments may be made to the implementation to guarantee performance prior to deployment of the system in an operational environment.

10   Embodiments of the invention also relate to improving the accuracy of predictive modeling of an information system. According to one embodiment, a system and method is disclosed for improving the accuracy of predictive modeling of an information system by modeling a dynamic representation of the business solution. According to a another embodiment, a system and method is disclosed for improving
15   the accuracy of predictive modeling of an information system through automated calibration of a predictive model 50 against predefined performance benchmarks.

As previously discussed, a business solution involves interactions among a number of business processes and business functions. Application and system components, in turn, support the business processes and functions. The business
20   layer 60 provides workload parameters which are injected into the system of equations 120 that represents the application and system layers 70, 80, while the probabilities of dynamic characteristics and behavior in the application and system layers 70, 80 are used to generate performance metrics 125 associated with the business layer 60. However, in prior art systems, the business layer 60 itself did not express dynamic
25   characteristics and behavior of the business components and their interactions. Thus, the performance metrics of the business layer were not fully representative of the dynamic nature of the business solution.

Embodiments of the invention provide a system and method for improving the accuracy of predictive modeling. New dynamic business characteristics and drivers in

the business layer 60 translate into new dynamic application characteristics and drivers in the application layer 70 for improved accuracy by capturing in more detail the characteristics and drivers of a business organization. The predictive model 50, in turn, converts the dynamic business characteristics and behavior into application and system

5   specific characteristics and drivers. For example, locks in a business process are represented as time-dependent characteristics in the application or component. Thus, a business solution is more accurately reflected in the predictive model 50 of the present invention.

For example, FIG. 11 is a diagram illustrating a prior art representation of a
10   business process design for a stock exchange. The primary business process 510 is a stock matching process through which buyers and sellers trade stock. The matching process 510 includes three subprocesses: a buyer verification process 520, a seller verification process 530, and a stock exchange process 540. Business workload 500 represents the volume of stock transaction requests (*e.g.*, 100 requests).

15   If the percentage of buyers and sellers is assumed to be 50%, then 50 buyer and 50 seller stock requests proceed to the verification processes 520 and 530. Assuming further that the percentage verified is 100%, 50 buyer and 50 seller stock requests are then forwarded to the stock exchange process 540. In prior systems, the stock exchange process 540 would result in 50 stock transactions being executed within the time
20   required by the technical architecture to process the transactions. In real deployable information systems, however, much less than 50 stock transactions would typically result in the time to process all the stock requests. Some transactions may occur immediately, while other transactions may take much longer.

For instance, if 20% of the buyer stock requests are for stock A and 10% of the
25   seller stock requests are for stock A, there would be 5 stock transactions executed immediately involving stock A with 5 outstanding buyer stock requests waiting to purchase stock A. Thus, even if the system architecture is running efficiently, the business response time is much longer and may indicate that the business process design is not sufficient to satisfy its business requirements.

In contrast, FIG. 12 is a diagram illustrating a dynamic business representation of a business solution according to one embodiment of the present invention. With a dynamic representation of a business solution, it may be determined from the performance metrics 125 whether the business solution can actually meet the business

5   needs, regardless of whether the performance metrics of the application and system layers 70, 80 are acceptable.

In generating the dynamic business representation, a description of a business solution is provided describing business components (collectively 610) and the interactions (collectively 620) among them. According to one embodiment, the

10   description includes dynamic drivers for the business processes or functions (sub-processes and sub-functions), such as, but not limited to: (i) number and kind of business events; (ii) probability to perform different sequences of sub-process/sub-functions; (iii) mode of use (*e.g.*, concurrent, sequential, differed, right time, seasonality, regularity/volatility, genericity/specificity, etc.); (iv) weight of an

15   event with respect to others (mix, stability, exception); (v) arrival rates in kind and value; (vi) arrival mechanism; and (vii) mode of chaining and access, which is translated into demands upon the infrastructure layers and services.

From the description of the business solution, a predictive model 50 is generated with the business layer 60 providing a dynamic representation of the business

20   solution. In particular, the business layer 60 models dynamic characteristics and behavior of the business components and the interactions among them in response to dynamic business workloads. According to one embodiment, the business layer 60 models the dynamic behavior and characteristics and the dynamic drivers for business processes, sub-processes, functions, and sub-functions. The dynamic behavior

25   characteristics include, but are not limited to, (i) process or function locks (*i.e.*, waiting for a business event), (ii) the control and management disciplines that guarantee the business events execution and integrity, and (iii) distribution into sub-processes and sub-functions with each representing a stage or step in performing the process or function). The distribution may be horizontal (*e.g.*, concurrent, parallel, or sequential)

or vertical (hierarchical chaining with the proper mechanism of connections and accesses that will be delivered from the infrastructure layers). Equation (1) is utilized to calculate the probabilities of the business layer 60 and its components.

5   Referring to FIG. 12, arrows 620 represent the dynamic characteristics and behavior of an interaction between business components. The interaction may be expressed with its own component model equation, $S_{j,m}$ within equation (1). In prior systems, delays were not modeled in interactions resulting in inaccurate performance metrics requiring manual calibration of the predictive model. According to one embodiment, the dynamic characteristics and behavior expressed in the model equation

10   for an interaction 620 may include one or more probabilities of delays. Such delay may be associated with conflicts, contentions, locks or further processing external to a business component.

     With respect to locks, a lock is the dependency of a business process or function on another business process or function, respectively, for processing or information

15   retrieval services. A lock is characterized by a delay representing the duration that a business process or function is locked, waiting for the requested processing and/or information. For example, in a securities exchange, a business function that calculates the cost of a stock transaction may depend on another function to provide real-time stock quotes used in its calculation. The calculation function may be prevented from

20   completing its calculation (*i.e.*, "locked") until the quoting function returns the requested stock quotes.

     By identifying locks in the business layer 60, the constituent components of the locks may be identified in the application and system domain and their effects may be modeled as time-dependent characteristics. With the ability to identify locks at the

25   business and application layers 60, 70, an information system designer may be able to consider alternative architectures, which avoid the locking behavior.

     As with any business, some business processes/functions rely on input from other processes/functions. Such requests for data/results is not instantaneous. Response times are dependent on a number of factors, such as outstanding requests to the

organization or department that provides such information, the amount of processing involved in providing the data/results requested, the applications and system infrastructure implemented to carry out its business process or function. Thus, although not intrinsic to a business process or function, communication exchanges between

5   business processes and business functions can affect performance metrics, such as response time and transacted business volume. Examples of such exchanges include information transfers, mergers, and extraction. Modeling of such exchanges provides a more accurate representation of the dynamics within the business management domain.

Continuing with FIG. 12, information transfers 623 between business processes

10   and functions, such as workload transfers, may be effectuated by a number of transmission mediums, such as facsimile, phone, mail delivery, hand delivery, or electronic transmission means (*e.g.*, email). For example, in a securities exchange, the transmission of orders from a brokerage firm to the securities exchange is a typical workload transfer. Some exchanges may additionally require authorization from a

15   customer or supervisor prior to transfer.

Mergers 625 are a particular type of interaction in which two or more interactions are merging content or workload in the same business component. In particular, information mergers typically involve updating information maintained in a storage system, such as a database. For example, a business process in a stock exchange

20   may continuously track changes in stock prices, merging updates into a database of stock quotations by adding, modifying, and deleting stock quotations. According to one embodiment, the dynamic characteristics and behavior of a merger includes a probability of a delay associated with completion of the merger.

Extractions 627 are another type of interaction in which business content is

25   retrieved from a business component, such as a data store. For example, a business function in a stock exchange may request real-time stock quotations, extracting the information from a stock quotation database. Such extractions are not instantaneous and depend on the amount of activity (*e.g.*, data queries) present on the business component. According to one embodiment, an interaction represents an extraction of

business content from a business component with the dynamic characteristics and behavior of the extraction including a probability of a delay associated with the extraction.

By modeling inter-process and inter-function exchanges, delays incurred by such

5   exchanges may be identified and modeled at one or more layers of each domain, providing a more accurate representation of the modeled information system.

In another embodiment, the dynamic characteristics and behavior of an interaction between business components may include one or more probabilities of business workload type being processed. In a further embodiment, the dynamic

10   characteristics and behavior of an interaction between business components may include one or more probabilities of an occurrence of one or more business events.

According to one embodiment, the dynamic characteristics and behavior of a business component may differ in response to business workload type or business event. Such differences may include execution sequences of business components. An

15   execution sequence of business components may also be affected by time constraints associated with a business event, such as right time constraints (e.g., event must be processed within 2 hours). Furthermore, execution sequences of business components may also be dependent on locks wherein execution stalls until a particular event occurs.

According to a further embodiment, the business layer 60 models dynamic

20   characteristics and behavior of business components having different modes of operation, such as batch processing, transactional processing, messaging, or query-based processing.

According to one embodiment, the business layer 60 models the distribution of business processes, vertically into information system model (including application

25   layer 70, system layer 80, and data stores) or horizontally into sub-processes, sub-functions, and interactions. In particular, business processes may be horizontally or vertically distributed over a number of business sub-processes in order to maximize usability and reduce complexity. Likewise, business functions may be horizontally or vertically distributed over a number of business sub-functions. Horizontal distribution

includes parallel, concurrent, and sequential distributions, while vertical distribution is a hierarchy of linked business sub-processes.

By identifying business process and function distributions in the business management domain, distribution-related constraints may be imposed on the application

5   and system domains. For example, distributions in the business management domain may constrain the application domain, such that groups of software application components are distributed horizontally across one or more servers or vertically into smaller, individual, and differentiated components. Similarly, the system domain may be constrained with respect to the hardware components available to support such

10   distributions (*e.g.*, parallel processors, networked servers).

Thus, embodiments of the invention provide a system and method for improving the accuracy of predictive modeling by generating a predictive model 50 of the information system including a business layer 60 generated from the business solution description that models dynamic characteristics and behavior of the business

15   components and the interactions among them in response to dynamic business workloads, such that a dynamic representation of the business solution results.

Thus, with a dynamic business representation, the performance metrics 125 of the business layer 60 may indicate whether the business solution satisfies a set of business requirements regardless of whether the performance metrics 125 of the

20   application and system layers 70, 80 are acceptable. If the performance metrics 125 from the business layer 60 indicate that the business solution would not satisfy the business requirements, then the business solution needs to be modified adding additional capacity, which, in turn, affects the design of the application and system layers 70, 80.

25   FIG. 13 is a flow diagram of a process for improving the accuracy of predictive modeling according to one embodiment, referred to as an accuracy evaluation 140 in FIG. 4.

At 710, a predictive model 50 of an information system is provided for an accuracy assessment. The accuracy assessment is typically performed during the

predictive modeling phase 260 for a design resulting from the technical architecture design phase 230.

At 720, individual performance metrics 125 are calculated for each component.

At 730, the calculated individual performance metrics 125 are compared against predefined performance benchmarks 115 associated with each component. The performance benchmarks 115 may be provided by the vendor of the component, measured, or observed through component testing.

At 740, for each component, if a calculated performance metric 125 for a component matches the performance benchmark 115 within a predefined threshold (*e.g.*, within 98% of the benchmark), the predictive model 50 of the component is accurate with respect to the performance benchmarks 115 and, thus, proceeds to a next development phase at 750. Alternatively, if the calculated performance metric 125 does not match the performance benchmark 115 within the threshold, the component model might be inaccurate and, thus, the process proceeds to 760 where the component model is calibrated such that it may calculate performance metrics 125 that meet the performance benchmarks 115.

At 760, the parameters of the equation modeling a dynamic characteristic or behavior of the component is adjusted. For example, assume that we have a CPU coupled to four disk drives. The performance metric, processor utilization, has a performance benchmark of 75%. The model input parameters 110 specify that over a one hour period, there should be 10 I/Os per disk. This may result in a processor utilization of 50%, which is not accurate. Thus, the parameters 10 I/Os per disk may be incorrect. Thus, step 760 redistributes the parameters such that there are 20 I/Os to Disk1, 10 I/Os to Disk2 and 5 I/Os per Disk3 and Disk4 (which still provides an average of 10 I/Os per disk). After reevaluating this revised or adjusted model 180, (iterating through steps 720, 730, and 740) the process 140 may find that the processor utilization is now 65%. This sensitivity analysis repeats until a processor utilization that matches the performance benchmark of 75% within a predefined threshold is obtained.

With embodiments of automated calibration of a predictive model 50 against performance benchmarks 115, the predictive model 50 may be utilized as a reference for calculating performance metrics 125 for different volumes and workload types. Thus, the inventive method and system provides a system designer with the ability to predict

5   whether a particular design may scale easily for dynamic changes in business volume and workload.

While this invention has been particularly shown and described with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the

10   scope of the invention encompassed by the appended claims.